Client's ref.: 02-0296
File: 0608-A40045US/final/mike

TITLE

METHOD OF WRITE-PROTECTING A MAC ADDRESS

BACKGROUND OF THE INVENTION

Field of the Invention

The present invention relates to a method of write-protecting a MAC address, and more particularly write-protecting of a MAC address of a peripheral terminal stored in DMI memory when the BIOS is updated.

Description of the Related Art

Although networks make many functions more effective, increasing use of computers and various peripheral devices create many problems for management. Thus, the desktop management interface (DMI) standard was established by the desktop management task force (DMTF). Presently, management information format (MIF) for the peripheral device, such as processor, BIOS, cache, LAN, and IEEE1394 peripheral devices, has been defined by the system standard definition groups thereof. Management information also comprises media access control (MAC) address for each peripheral terminal.

Fig.1 is a schematic diagram of a partial structure of a system on a motherboard. A CPU 10, a Northbridge chip 11 connected to the CPU 10, a Southbridge chip 12 connected to the Northbridge chip 11, a LAN interface 120, and an electrically erasable programmable read-only memory " EEPROM " 121 are installed on a motherboard. The MAC address is generally stored in the EEPROM. However, the memory generates extra costs.

Another conventional method is shown in Figs.2 and 3. Fig.2 shows a system connected to a local area network (LAN) 20, an

IEEE1394 peripheral device 21 and another peripheral terminal 22 on a motherboard 23. The motherboard 23 comprises a Southbridge chip 200, a Northbridge chip 201, a BIOS memory 202, and a CPU 203. The DMI data is stored in the BIOS memory 202.

Fig.3 is a table showing the storage space of the BIOS memory. The DMI data comprises LAN DMI data 30, IEEE1394 peripheral device DMI data 31, and other DMI data 32 of other peripheral devices. Each of DMI data 30, 31, and 32 comprises a MAC address 33 and a slot identification (slot ID) code 34.

However, by downloading a utility program such as FLASH.EXE, DMICFG, the DMI data can be erased or modified by changing parameters and executing the downloaded program. Therefore, the MAC address may be overwritten when updating the data, causing system malfunction.

SUMMARY OF THE INVENTION

Accordingly, an object of the present invention is to prevent MAC address of a peripheral terminal stored in a DMI memory from being overwritten.

In order to achieve the above object, the invention provides a method of write-protecting a MAC address of a peripheral terminal stored in a DMI memory.

First, programs capable of erasing the MAC address stored in the first memory are disabled. Then, a DMI setting is executed to write-protect the MAC address stored in the second memory. Finally, a program is provided capable of pre-storing the original MAC address.

A detailed description is given in the following with reference to the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention can be more fully understood by reading the subsequent detailed description and examples with references made to the accompanying drawings, wherein:

Fig.1 is a block diagram of a computer system;

Fig.2 is a block diagram of a computer system connected to peripheral devices;

Fig.3 is a table showing conventional storage space of a BIOS memory;

Fig.4 is a flowchart of a method of write-protecting a MAC address of a peripheral terminal stored in a DMI memory according to the present invention;

Fig.5 is a flowchart of the detailed process of step 40 in Fig.4; and

Fig.6 is another flowchart of a method of write-protecting a MAC address of a peripheral terminal stored in a DMI memory according to the present invention.

DETAILED DESCRIPTION OF THE INVENTION

Fig.4 is a flowchart of a method of write-protecting a MAC address of a peripheral terminal stored in a DMI memory according to the present invention. This method prevents the MAC address from being erased, providing a function capable of pre-storing the original MAC data, wherein the MAC address is stored in a first memory (hereafter CMOS memory), and a backup MAC address is stored in a second memory (hereafter DMI Flash Memory), wherein the second memory is a non-volatile memory. The method comprises the following steps.

In step 40, programs capable of erasing the MAC address stored in the CMOS memory are disabled. In step 41, a DMI setting

prevents the MAC address stored in the DMI Flash Memory from being overwritten. Finally, in step 42, a program is provided capable of pre-storing the original MAC address stored in a predetermined register to further back up the MAC address, such that, if the original MAC address is erased, the original MAC address can be recovered.

Fig.5 is a flowchart of the detailed process of step 40 in Fig.4, comprising the following steps.

First, in step 400, a function capable of limiting the execution of an AWDFLASH.EXE program is provided. Next, in step 401, a function capable of limiting the writing of a DMICFG.EXE program is provided. Because the programs AWDFLASH.EXE and DMICFG.EXE are limited, parameters of the DMI data cannot be erased or modified.

Fig.6 is a flowchart according to another embodiment of the present invention. This embodiment is a process for performing a subprogram of a power-on self test (POST) program, and comprises the following steps.

First, in step 50, the process of a subprogram of the POST program is started to determine whether the checksum value (identification code) of the MAC address stored in the DMI Flash memory is correct. If so, it is determined whether the checksum value of the MAC address stored in the CMOS memory is correct in step 51. If so, the MAC address stored in the DMI memory is copied to a shadow register in the LAN card in step 53. Next, the setting function of the MAC address is hidden in step 54. Namely, the setting function of the CMOS MAC address in the setup frame is not displayed. If the MAC address stored in the CMOS memory is incorrect, the MAC address stored in the DMI flash memory is copied to the MAC address in the CMOS memory in step

52, and steps 53 and 54 are performed, which, having been described, are not repeated here.

In step 50, if the checksum value stored in the MAC address in the DMI flash memory is incorrect, it is determined whether the checksum value of the MAC address in the DMI is equal to B1h in step 55. If so, the flag of the updated MAC address of the DMI is set in step 56, then steps 52 to 54 are performed. If the checksum value of the MAC address in the DMI is not equal to B1h, it is determined whether the checksum value of the MAC stored in the CMOS memory is correct in step 57. If so, the MAC address in the CMOS memory is copied to the MAC address in the DMI flash memory in step 58. Steps 56 and 52 to 54 are performed.

While the invention has been described by way of example and in terms of the preferred embodiment, it is to be understood that the invention is not limited to the disclosed embodiments. On the contrary, it is intended to cover various modifications and similar arrangements as would be apparent to those skilled in the art. Therefore, the scope of the appended claims should be accorded the broadest interpretation to encompass all such modifications and similar arrangements.